1 Problem 1

1. *In this and the following exercises, you are asked to design a Kalman filter for a simple dynamical system: a car with linear dynamics moving in a linear environment. Assume $\delta t = 1$ for simplicity. The position of the car at time t is given by x_t. Its velocity is given by $\dot{x}_t$, and its acceleration [from time $t-1$ to time t] is given by $\ddot{x}_t$. Suppose the acceleration is set randomly at each point in time, according to a Gaussian with zero mean and covariance $\sigma^2 = 1$.*

   One thing I note: At first it sounded to me like we would know what the acceleration value was at each time step. This threw me off for a long time, until I realized that this is more like modeling a random wind blowing - you don't know what the effect is at any given moment, all you know is it's statistical behavior. Since we don't *know* the value of the acceleration, it *can't* be part of our state! That may be obvious to others reading the problem, but I worked on it for two days (!) before I figured it out. C'est la vie.

   (a) *What is a minimal state vector for the Kalman filter (so that the resulting system is Markovian)?*

   In order to be Markovian, we have to have a state vector such that the future and the past are independent given the present state. If the state vector has the position and velocity, this condition is met. All of the acceleration values of the past are completely summarized in the position and velocity, so keeping it as a state variable doesn't tell us anything new in terms of predicting the future. The acceleration is set randomly at each time step, but given that we know the position and velocity we don't need to know any past acceleration in order to compute the future given the state and the control (which in this case will be setting the acceleration).

   Our state vector is thus:

   $$\begin{pmatrix} x_t \\ \dot{x}_t \end{pmatrix}$$

   (b) *For your state vector, design the state transition probability $p(x_t|u_t, x_{t-1})$. Hint: this transition function will possess linear matrices A and B and a noise covariance R.*

   We will use the moments parameterization since we're building a Kalman filter. The state transition function is of the form

   $$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

   Where ϵ_t is a gaussian random variable with mean zero and variance 1 (the acceleration, as given in the problem). We'll assume there is no acceleration, and incorporate the acceleration as error in the position and velocity.

The mean for our distribution is given by $Ax_{t-1} + Bu_t$, and the variance is given by $A\Sigma_{t-1}A^T + cov(\epsilon_t)$. The $A\Sigma_{t-1}A^T$ term carries forward the uncertainty from the previous state updated by the state transition function (i.e. we need to apply the same transformation to our error bounds that we apply to the estimate, otherwise the error bounds will become meaningless). The $cov(\epsilon_t)$ term is the new error added in by the random acceleration.

All that remains is to use the equations of motion to derive matrices for A and B. We know that:

$$x_t = x_{t-1} + \left(\frac{\dot{x}_{t-1} + \dot{x}_t}{2}\right)\Delta_t \tag{1}$$

$$\dot{x}_t = \dot{x}_{t-1} + \ddot{x}_t\Delta_t \tag{2}$$

Using our expression for $\dot{x}_t$ in the first equation yields:

$$x_t = x_{t-1} + \dot{x}_t - 1\Delta_t + \frac{1}{2}\ddot{x}_t\Delta_t^2 \tag{3}$$

We use the average velocity between the start and end of the time slice to calculate the new x_t value, because the velocity has changed in the interval as a result of the acceleration ($\ddot{x}_t$).

There is no control action in this model, since there is no explicit (known) acceleration (i.e. the acceleration value is just a probability distribution and we don't know its value at any given time). That means $Bu_t = \vec{0}$.

Now we can write our prediction of the overall mean, in matrix form:

$$\overline{\mu}_t = \begin{pmatrix} x_t \\ \dot{x}_t \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{pmatrix}$$

Next we need to model the covariance. Since we are given the variance of the acceleration (=1), we have to figure out how that maps into variance of the two state variables and make a variance vector accordingly.

From our equations of motion, we know that x_t depends on $\ddot{x}_t$ scaled by $\frac{\Delta_t}{2}$, and we know that $\dot{x}_t$ depends on $\ddot{x}_t$ without scaling (i.e. the scale factor is 1).

That gives us what we need to figure out how the random acceleration should affect our estimates for position and velocity. The variance of velocity should be equal to the variance of acceleration times the time elapsed, and the variance of position should be equal to half of the squared time elapsed.

If we knew what the acceleration was at each step, we could add a correction term to the above expression for the mean to get an exact answer. This is what that term would look like:

$$\delta_t = \begin{pmatrix} \frac{\Delta_t^2}{2} \\ \Delta_t \end{pmatrix} \ddot{x}_t \tag{4}$$

However, we don't have knowledge of $\ddot{x}_t$, all we know is that it's gaussian with mean zero and std deviation 1. We therefore need to turn that equation into a multivariate gaussian distribution, which when added to our mean will yield the total probability distribution for x_t. In order to do that, we need to calculate a covariance matrix based on the vector δ_t (which is telling us how much the acceleration impacts each of the state variables).

We know the variance of the acceleration, but we need to figure out how that maps into the variance of the position and velocity. Our δ vector gives us the scaling factors - if we know the standard deviation of $\ddot{x}_t$, then we could scale it by the values in δ_t to see how it would change the standard deviation of x_t and $\dot{x}_t$. However, since we're working with a gaussian distribution, we want to use variance. That means that we're working with the squared standard deviation, so we need to apply the appropriate analogous operation to δ_t in order to make a covariance matrix. We can do that by calculating $\delta_t \delta_t^T$ (yes, I'm hand-waving a bit here because I haven't figured out the intuitive explanation for *why* this should make the right covariance matrix yet...):

$$R = cov\left(\delta_t\right) = \delta_t \delta_t^T = \begin{pmatrix} \frac{\Delta_t^2}{2} \\ \Delta_t \end{pmatrix} \begin{pmatrix} \frac{\Delta_t^2}{2} & \Delta_t \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 \end{pmatrix} \tag{6}$$

Now we have our probability $p(x_t|u_t, x_{t-1})$:

$$\overline{\mu}_t = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{pmatrix} \tag{7}$$

$$\overline{\Sigma}_t = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix} \Sigma_{t-1} \begin{pmatrix} 1 & 0 \\ \Delta_t & 1 \end{pmatrix} + \begin{pmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 \end{pmatrix} \sigma_{\ddot{x}_t}^2 \tag{8}$$

We have to provide an initial value for Σ_0. In this case, I think it's reasonable to assume that it is 0. The uncertainty from the random accelerations at each timestep will propagate into Σ_t, so it will not remain 0 for long.

Later, the Kalman gain will be used to select how much to weight the prediction versus the correction (measurement), based on the relative magnitudes of the covariances.

(c) *Implement the state prediction step of the Kalman filter. Assuming we know at time* $t = x, x_0 = \dot{x}_0 = \ddot{x}_0 = 0$. *Compute the state distributions for times* $t = 1, 2, ..., 5$.

(d) *For each value of* t*, plot the joint posterior over* x *and* $\dot{x}$ *in a diagram, where* x *is the horizontal and* $\dot{x}$ *is the vertical axis. For each posterior, your are asked to plot an uncertainty ellipse, which is the ellipse of points that are one standard deviation away from the mean. Hint: if you do not have access to a mathematics library, you can create those ellipses by analyzing the eigenvalues of the covariance matrix.*

(e) *What will happen to the correlation between* x_t *and* $\dot{x}_t$ *as* $t \to \infty$*?*

2. *In Chapter 3.2.4, we derived the prediction step of the KF. This step is often derived with Z transforms or Fourier transforms, using the Convolution theorem. Re-derive the prediction step using transforms.*

   I didn't do this one, it seems like a lot more work than it's worth for my goals. Maybe I'll come back to it...

3. *We noted in the text that the EKF linearization is an approximation. To see how bad this approximation is, we ask you to work out an example. Suppose we have a mobile robot operating in a planar environment. Its state is its x-y location and its global headiing* θ*. Suppose we know* x *and* y *with high certainty, but the orientation* θ *is unknown. This is reflected in our initial estimate:*

$$\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{9}$$

$$\Sigma = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 10000 \end{pmatrix} \tag{10}$$

(a) *Draw, graphically, your best model of the posterior over the robot pose after the robot moves* $d = 1$ *units forward. For this exercise, we assume that the robot moves flawlessly without any noise. Thus, the expected location of the robot after motion will be:*

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + cos\theta \\ y + sin\theta \\ \theta \end{pmatrix} \tag{11}$$

(b) *Now develop this motion into a prediction step for the EKF. For that, you have to generate a new Gaussian estimate of the robot pose using the linearized model. You should give the exact mathematical equations for each of these steps, and state the Gaussian that results.*

(c) *Draw the uncertainty ellipse of the Gaussian and compare it with your intuitive solution.*

(d) *Now incorporate a measurement. Our measurement shall be a noisy projection of the x-coordinate of the robot, with covariance* $Q = 0.01$*. Specify the measurement model. Now apply the measurement both to your intuitive posterior, and formally to the EKF estimate using the standard EKF machinery. Give the exact result of the EKF, and compare it with the result of your intuitive analysis.*

(e) *Discuss the difference between your estimate of the posterior, and the Gaussian produced by the EKF. How significant are those differences? What can be changed to make the approximation more accurate? What would have happened if the initial orientation had been known, but not the robot's y-coordinate?*

4. *The Kalman filter in Table 3.1 lacked a constant additive term in the motion and the measurement models. Extend this algorithm to contain such terms.*

5. *Prove (via example) the existence of a sparse information matrix in multivariate Gaussians (of dimension d) that correlate all d variables with correlation coefficients that are* ϵ*-close to 1. We say an infomration matrix is sparse if all but a constant number of elements in each row and column are zero.*